Rank-Codes for OFDM

Simon Plass
German Aerospace Center (DLR)

4th MCM of COST 289
Zurich, March 15, 2004

Overview

- Motivation
- Introduction of Rank-Codes
- Algebraic decoding
- Berlekamp-Massey algorithm for Rank-Codes
- OFDM
- Simulation results
- Conclusions & outlook

Motivation

- Signals of multi-carrier transmission systems can be represented in a matrix form:

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,n} \end{pmatrix}$$

frequency

time

- Rank-Codes are defined in a matrix form and decode over a matrix.

➔How can Rank-Codes be implemented?

➔May a combination of multi-carrier systems and Rank-Codes result in a good performance?

Introduction of Rank-Codes

Let us consider a vector with elements of the extension field *GF(q^N)*:

$$x = (x_1, x_2, \ldots, x_n)$$

Now, we can present the vector *x* as a matrix with entries of the finite field *GF(q)*:

$$A(x) = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,1} & \cdots & a_{N,n} \end{pmatrix}$$

Let us define the rank distance between two matrices A and B as:

$$d_r(A,B) = rank(A-B)$$

Introduction of Rank-Codes (cont'd)

Example for the rank distance:

$$d_r(A,B) = rank\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\right) = rank\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\right) = 2$$

Furthermore, Rank-Codes have an error correction capability *t* of

$$rank(E) \le t = \left\lfloor \frac{d_r - 1}{2} \right\rfloor$$

where E is the error matrix.

Example of Rank Error

0	0	0	1	0
0	0	0	0	0
0	1	0	1	0
0	0	0	1	0
0	0	0	1	0

1 = error

0	0	0	1	0
0	0	0	0	0
1	1	1	1	1
0	0	0	1	0
0	0	0	1	0

Rank array is 2.
➔rank error = 2

Rank of array is still 2.

Construction of Rank-Codes

A parity-check matrix H and its corresponding generator matrix G which define the Rank-Code are given by:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_n \\ h_1^q & h_2^q & h_3^q & \cdots & h_n^q \\ h_1^{q^2} & h_2^{q^2} & h_3^{q^2} & \cdots & h_n^{q^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_1^{q^{d-2}} & h_2^{q^{d-2}} & h_3^{q^{d-2}} & \cdots & h_n^{q^{d-2}} \end{bmatrix} \qquad G = \begin{bmatrix} g_1 & g_2 & g_3 & \cdots & g_n \\ g_1^q & g_2^q & g_3^q & \cdots & g_n^q \\ g_1^{q^2} & g_2^{q^2} & g_3^{q^2} & \cdots & g_n^{q^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & g_3^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{bmatrix}$$

The elements $h_1, h_2, \ldots, h_n \in GF(q^N)$ and $g_1, g_2, \ldots, g_n \in GF(q^N)$

must be linearly independent over $GF(q^N)$.

Algebraic Decoding

Syndrome calculation $s=(c+e)H^T=eH^T$
Syndrome includes information of column and row values.
➔ **Key equation**

↓

Use of efficient algorithm,
e.g., **Berlekamp-Massey algorithm**,
for solving the system of linear equations
➔ Error polynomial

↓

Error value and error location computation
by recursive calculation
➔ Error vector e

↓

$c_{decode} = r - e$

Key Equation of Rank-Codes

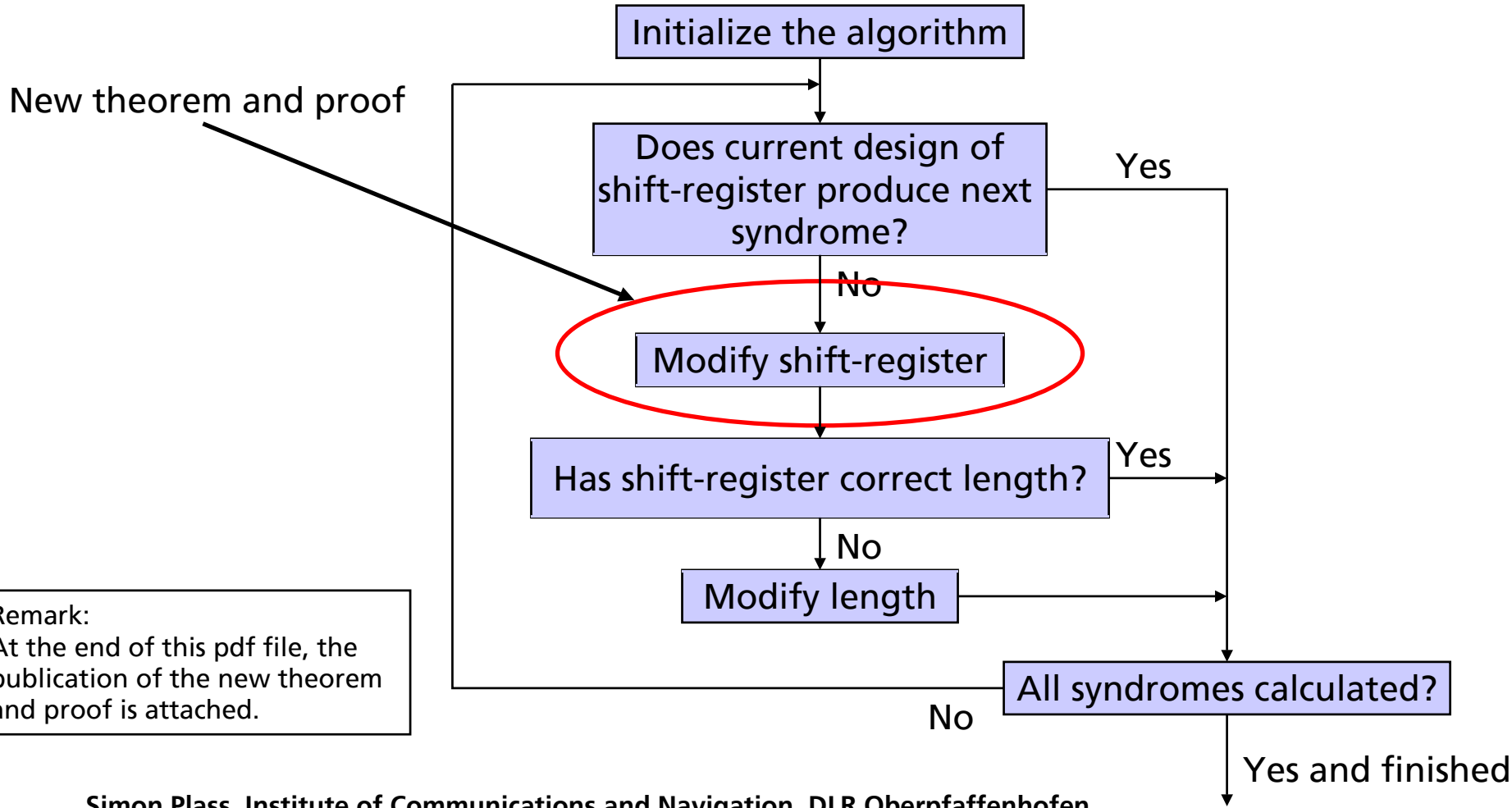
$$\begin{bmatrix} S_0^{q^\nu} & \cdots & S_{\nu-1}^{q^1} \\ S_1^{q^\nu} & \cdots & S_{\nu}^{q^1} \\ S_2^{q^\nu} & \cdots & S_{\nu+1}^{q^1} \\ \vdots & \ddots & \vdots \\ S_{\nu-1}^{q^\nu} & \cdots & S_{2\nu-2}^{q^1} \end{bmatrix} \begin{bmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \Lambda_{\nu-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_\nu \\ -S_{\nu+1} \\ -S_{\nu+2} \\ \vdots \\ -S_{2\nu-1} \end{bmatrix}$$

Syndrome S_j can be represented by an appropriate designed shift-register if Λ_i is known

$$S_j = -\sum_{i=1}^{\nu} \Lambda_i S_{j-i}^{q^i}, \qquad j = \nu, \ldots, 2\nu - 1$$

Main problem: Solve the key equation for the unknown variables Λ_i.

Berlekamp-Massey Algorithm for Rank-Codes



Remark:
At the end of this pdf file, the publication of the new theorem and proof is attached.

Algebraic Decoding

Syndrome calculation $s=rH^T=eH^T$
Syndrome includes information of column and row values.
➔ **Key equation**

Use of efficient algorithm,
e.g., **Berlekamp-Massey algorithm**,
for solving the system of linear equations
➔ Error polynomial

Error value and error location computation
by recursive calculation
➔ Error vector e

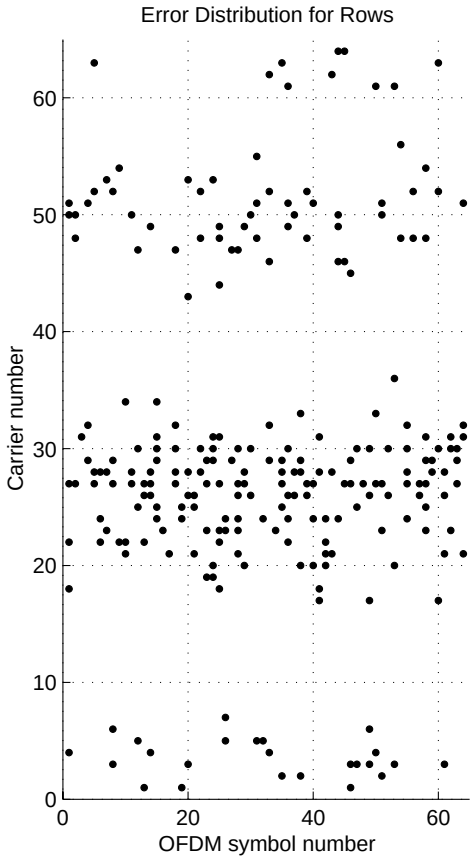
$c_{decode} = r - e$

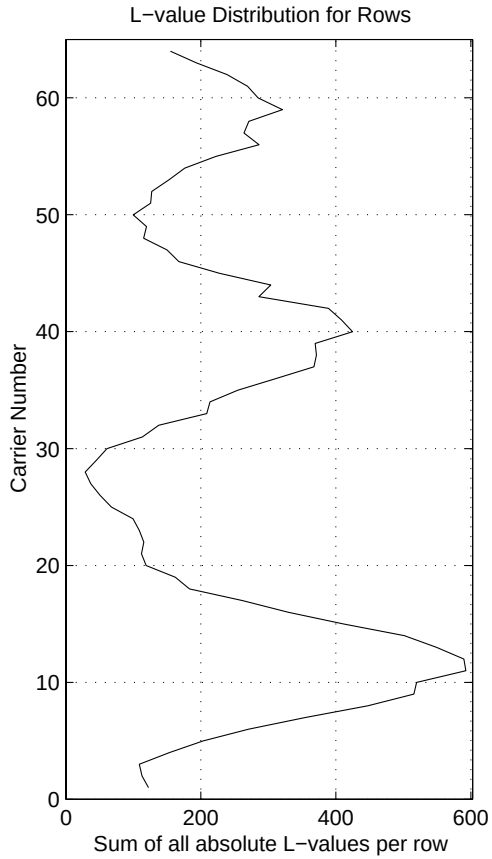
Orthogonal Frequency Division Multiplexing (OFDM)

- Benefits of OFDM are:
  - High spectral efficiency by using orthogonal signals (spectral overlapping).
  - Resistant against multi-delay spread by dividing the input data stream in *N* sub-carriers.

- Basic OFDM system parameters:
  - Indoor tapped delay-line multi-path channel model
  - Carrier frequency f_c=5.15GHz
  - BPSK modulation
  - Number of OFDM symbols = number of sub-carriers

Error Pattern of Transmission





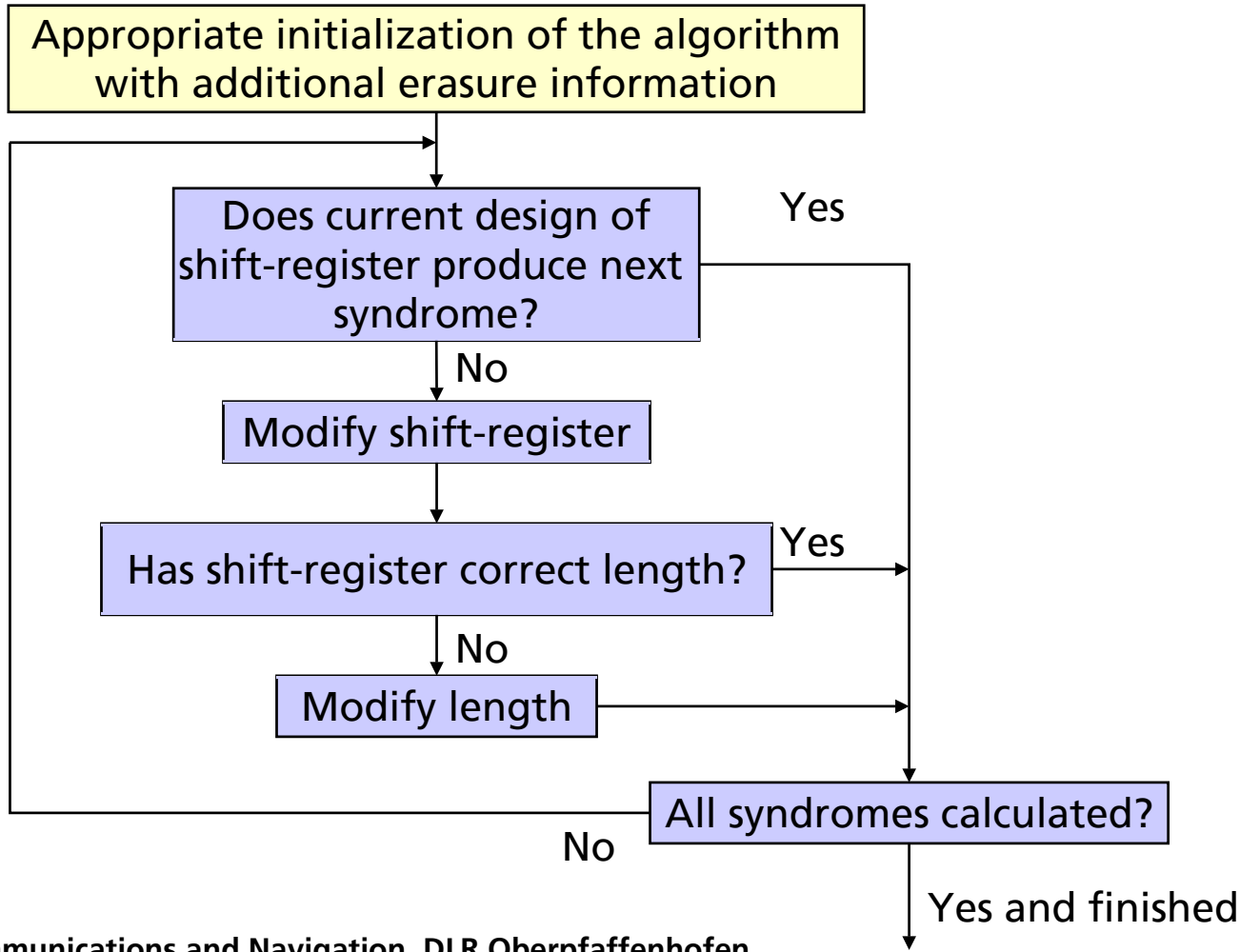
SNR=2dB, v=0km/h, n=64, BPSK

Rank-Code can decode

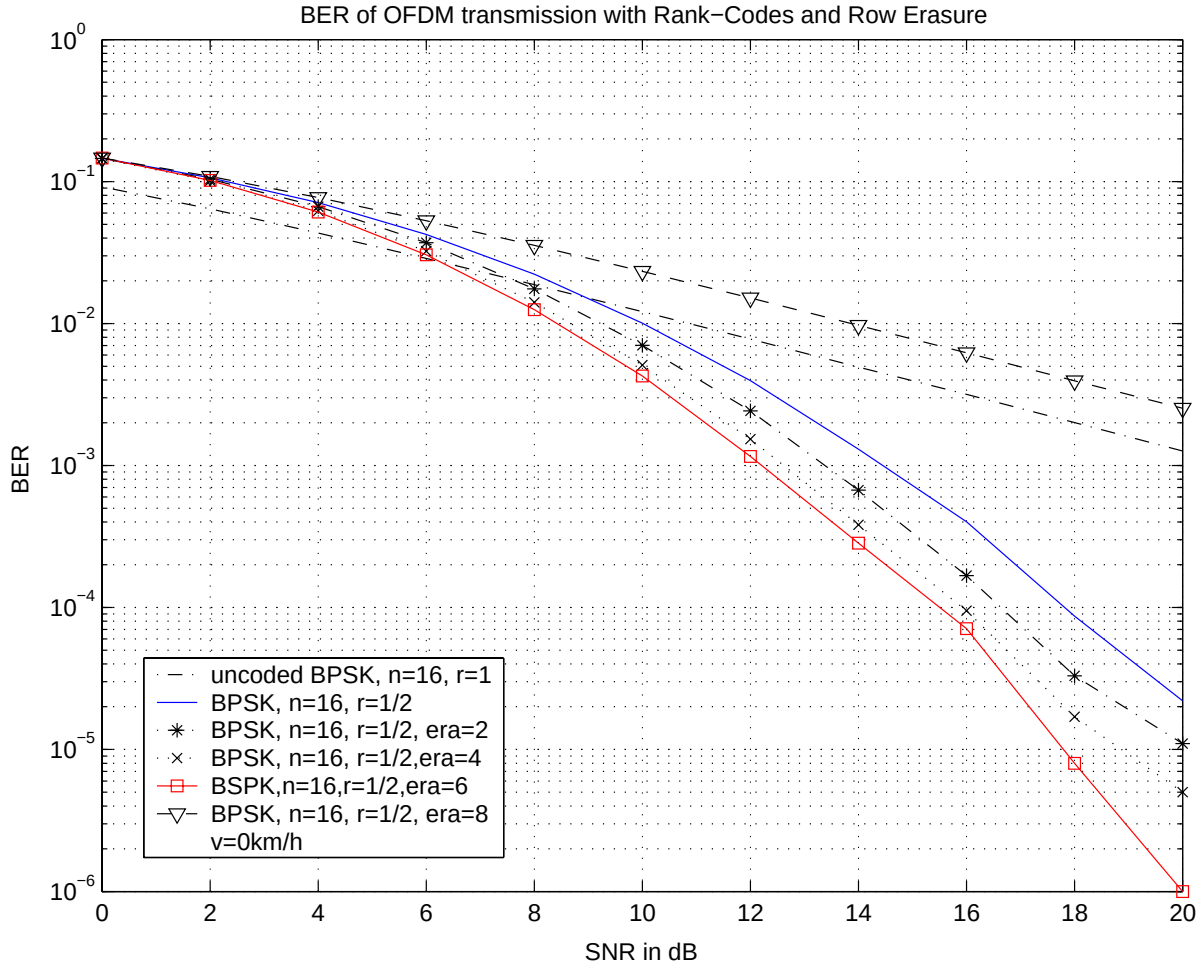
$$s + 2\nu < d_r$$

s is # of erasures
ν is # of errors

Berlekamp-Massey Algorithm for Rank-Codes with Row Erasures

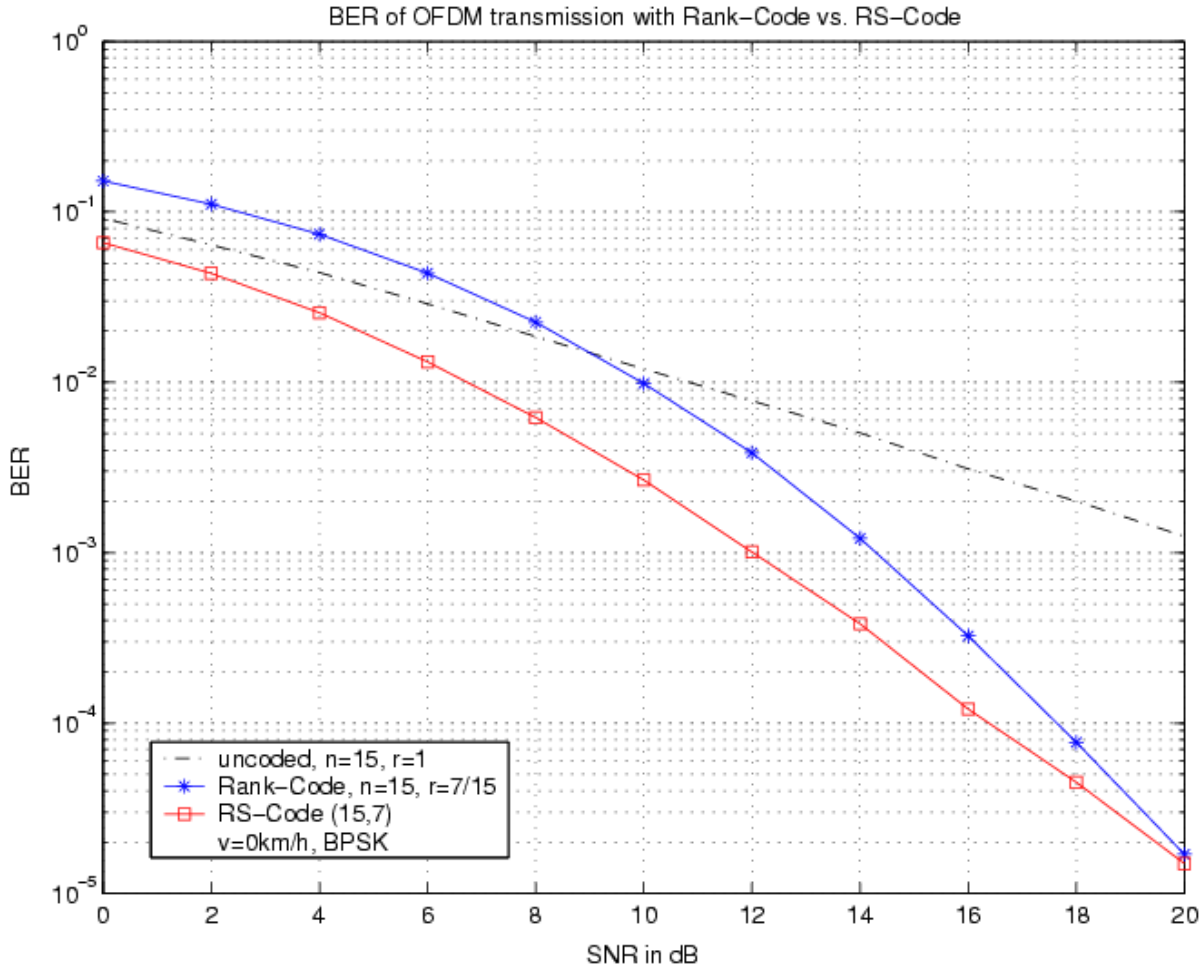


Performance of Rank-Codes with Row Erasure



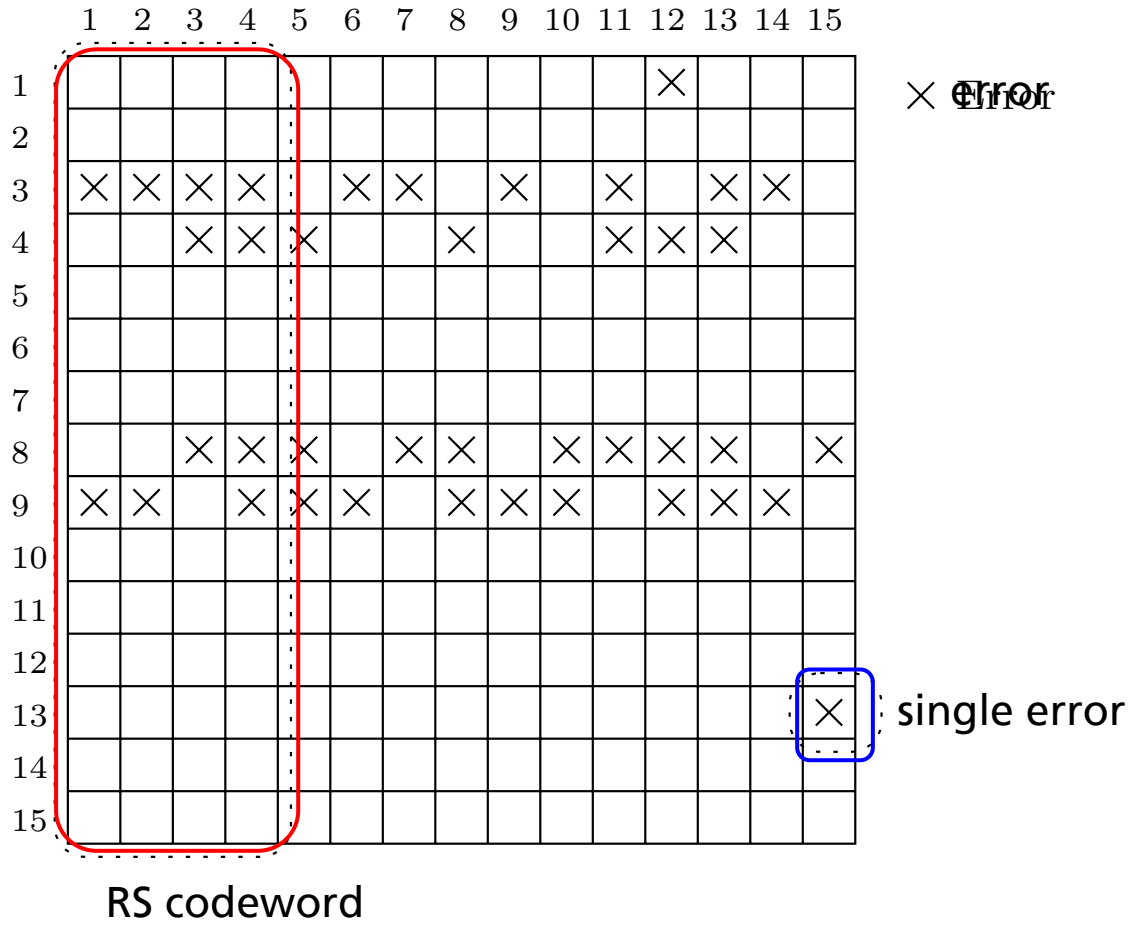
Simon Plass, Institute of Communications and Navigation, DLR Oberpfaffenhofen

Rank-Code vs. Reed-Solomon Code



Simon Plass, Institute of Communications and Navigation, DLR Oberpfaffenhofen

Error Pattern Example



Conclusions

- Use of an OFDM transmission model with Rank-Codes
- New approach of decoding algorithms for Rank-Codes:
  - -Modified Berlekamp-Massey algorithm for rank error correction
  - -Modified Berlekamp-Massey algorithm for row erasure and rank error correction
- OFDM channel characteristics are not optimal for Rank-Codes decoding properties

Outlook

- Implementation of Rank-Codes in systems with crisscross error patterns (e.g., memory chip arrays, magnetic tape recording)
- Including rank erasures for the Berlekamp-Massey algorithm

References

- G. Richter and S. Plass. Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm. *Int. ITG Conference on Source and Channel Coding*, pages 249-257, Erlangen, Germany, January 2004.
- G. Richter and S. Plass. Fast decoding of Rank-Codes with rank errors and column erasures. *IEEE Int. Symposium on Information Theory (ISIT 2004)*, Chicago, Illinois, USA, June 2004, accepted for publication.

Error and Erasure Decoding of Rank-Codes with a Modified Berlekamp-Massey Algorithm

Gerd Richter, Simon Plass
Dept. of Telecommunications and Applied Information Theory, TAIT
University of Ulm, Albert-Einstein-Allee 43, D-89081 Ulm, Germany.
E-mail: gerd.richter@e-technik.uni-ulm.de, simon.plass@web.de

Abstract

This paper investigates error and erasure decoding methods for codes with maximum rank distance. These codes can be used for correcting column and row errors and erasures in an $(N \times n)$ array. Such errors occur e.g. in magnetic tape recording or in memory chip arrays. For maximum rank distance codes (Rank-Codes), there exists a decoding algorithm similar to the Peterson-Gorenstein-Zierler technique for Reed-Solomon codes and a decoding method based on Euclid's Division algorithm. In this paper we introduce a modified Berlekamp-Massey algorithm for the decoding of rank errors and extend it for row erasures and rank errors. Also, we investigate a decoding algorithm for both row and column erasures and rank errors.

1 Introduction

In a number of applications, the following error protection problem occurs: The information symbols have to be stored in $(N \times n)$ arrays. Some of these symbols are recorded erroneously in such a way that all corrupted symbols are confined to a specified number of rows or columns (or both). We refer to such errors as crisscross errors. Figure 1 shows a crisscross error pattern that is limited to two columns and three rows.

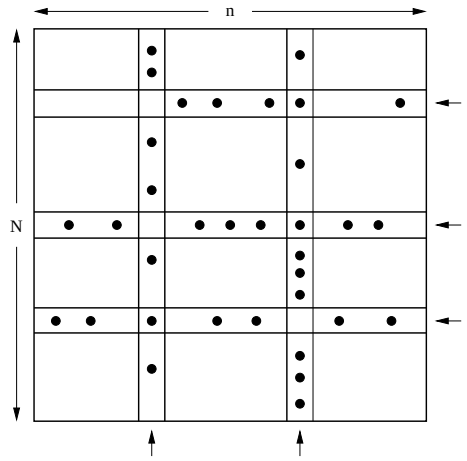


Fig. 1. Crisscross error pattern

These crisscross errors can be found in memory chip arrays [1], [2], [3] or in magnetic tape recording [4], [5], [6]. Since the Hamming metric is not appropriate for these error patterns, Delsarte [7] introduced the rank as a metric for error correction purpose. Gabidulin [8] and also Roth [9] introduced codes with maximum rank distance (Rank-Codes) that are capable of correcting a specified number of corrupted rows and columns. Rank-Codes cannot only correct erroneous rows and columns, they can even correct a certain number of rank errors. The number of rank errors is defined as the rank of the error array. This is shown in the following example.

Example 1 Assume that a Rank-Code can correct two rank errors. Let $\mathbf{R} = \mathbf{C} + \mathbf{E}$, where $\mathbf{R}$, $\mathbf{C}$, and $\mathbf{E}$ are the received array, the codeword array, and the error array, respectively. Let $q = 2$ and assume that $\mathbf{E}$ is given by:

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

It is easy to see that one column and two rows are corrupted. Anyhow, the Rank-Code can correct this error array because the rank of $\mathbf{E}$ is only two. ◇

There exist different algorithms for the decoding of Rank-Codes. Gabidulin [8] introduced the decoding with Euclid's Division algorithm based on linearised polynomials. In 1991, Roth has given another decoding algorithm [9] that is similar to the Peterson-Gorenstein-Zierler algorithm for Reed-Solomon codes.

These decoding algorithms are suitable for error correction. However, sometimes it is known which row or column is corrupted. We will call this a row or column erasure, respectively. The problem of decoding is now to find the correct values for the erased rows or columns and to correct the rank errors. With the

knowledge of erasures, the number of erroneous rows and columns that can be decoded increases. A method for erasure decoding was described by Gabidulin *et al.* in [10].

In 1968 Berlekamp [11] introduced a very efficient technique for the decoding of Reed-Solomon codes. One year later Massey [12] interpreted this algorithm as a problem of synthesising the shortest linear feedback shift-register capable of generating a prescribed finite sequence of digits. Since the structure of Reed-Solomon codes is quite similar to the structure of Rank-Codes, another possible decoding method is a modified Berlekamp-Massey algorithm which is introduced in this paper.

In section 2 we describe some important properties and a construction method of Rank-Codes. In section 3 we introduce a modified Berlekamp-Massey algorithm to correct rank errors. Section 4 deals with the decoding of row erasures and rank errors. Therefore, we extend the modified Berlekamp-Massey algorithm. The most difficult situation arises, when we want to decode row and column erasures (crisscross erasures) and rank errors. For this case we introduce a decoding algorithm similar to [10] in section 5.

2 Fundamentals of Rank-Codes

In this section we describe some fundamentals of Rank-Codes that were introduced by Gabidulin in 1985 [8]. Let $\mathbf{x}$ be a codeword of length n with elements from $GF(q^N)$, where q is a power of a prime. Let us consider a bijective mapping

$$\mathcal{A} : GF(q^N)^n \rightarrow \mathbf{A}_N^n$$

which maps the codeword $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ to an $(N \times n)$ array. In the following we consider only codewords of length $n \leq N$.

Definition 1 (Rank Metric over $GF(q)$) *The rank of $\mathbf{x}$ over q is defined as $r(\mathbf{x}|q) = r(\mathbf{A}|q)$. The rank function $r(\mathbf{A}|q)$ is equal to the maximum number of linearly independent rows or columns of $\mathbf{A}$ over $GF(q)$.*

It is well known that the rank function defines a norm. Indeed, $r(\mathbf{x}|q) \geq 0, \quad r(\mathbf{x}|q) = 0 \Longleftrightarrow \mathbf{x} = 0$. In addition, $r(\mathbf{x}+\mathbf{y}|q) \leq r(\mathbf{x}|q)+r(\mathbf{y}|q)$. Furthermore, $r(a\mathbf{x}|q) = |a|r(\mathbf{x}|q)$ is also fulfilled, if we set $|a| = 0$ for $a = 0$ and $|a| = 1$ for $a \neq 0$.

Definition 2 (Rank Distance) *Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords of length n with elements from $GF(q^N)$. The rank distance is defined as $\mathrm{dist}_r(\mathbf{x}, \mathbf{y}) = r(\mathbf{x}-\mathbf{y}|q)$.*

Similar to the minimum Hamming distance, we can determine the minimum rank distance of a code $\mathcal{C}$.

Definition 3 (Minimum Rank Distance) *For a code $\mathcal{C}$ the minimum rank distance is given by:*

$$d_r := \min\{\mathrm{dist}_r(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathcal{C}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\},$$

or when the code is linear:

$$d_r := \min\{r(\mathbf{x}|q) | \mathbf{x} \in \mathcal{C}, \mathbf{x} \neq 0\}.$$

Let $\mathcal{C}(n, k, d_r)$ be a code of dimension k, length n, and minimum rank distance d_r.

It was shown in [8] that there also exists a Singleton-style bound for the rank distance. Theorem 1 shows, how the minimum rank distance d_r is bounded by the minimum Hamming distance d_h and by the Singleton bound.

Theorem 1 (Singleton-style Bound) For every linear code $\mathcal{C}(n, k, d_r) \subset GF(q^N)^n$ d_r is upper bounded by:

$$d_r \leq d_h \leq n - k + 1.$$

Definition 4 (MRD Code) *A linear (n, k, d_r) code $\mathcal{C}$ is called **M**aximum **R**ank **D**istance (MRD) code, if the Singleton-style bound is fulfilled with equality.*

In [8] and [9] a construction method for the parity-check matrix and the generator matrix of an MRD code are given as follows:

Theorem 2 (Construction of MRD Codes)
A parity-check matrix $\mathbf{H}$ which defines an MRD code is given by:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0^q & h_1^q & \cdots & h_{n-1}^q \\ h_0^{q^2} & h_1^{q^2} & \cdots & h_{n-1}^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ h_0^{q^{d-2}} & h_1^{q^{d-2}} & \cdots & h_{n-1}^{q^{d-2}} \end{bmatrix}$$

and the corresponding generator matrix can be written as:

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^q & g_1^q & \cdots & g_{n-1}^q \\ g_0^{q^2} & g_1^{q^2} & \cdots & g_{n-1}^{q^2} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{q^{k-1}} & g_1^{q^{k-1}} & \cdots & g_{n-1}^{q^{k-1}} \end{bmatrix},$$

where the elements $h_0, h_1, \ldots, h_{n-1} \in GF(q^N)$ and $g_0, g_1, \ldots, g_{n-1} \in GF(q^N)$ are linearly independent over $GF(q)$.

In the following we define $\mathcal{C}_{\mathcal{MRD}}(n, k, d_r)$ as an MRD array code of length n, dimension k, and minimum rank distance $d_r = n - k + 1$.

The decoding of Rank-Codes with the modified Berlekamp-Massey algorithm can be done based on linearised polynomials.

Definition 5 (Linearised Polynomials) *A linearised polynomial over $GF(q^N)$ is a polynomial of the form*

$$L(x) = \sum_{p=0}^{N(L)} L_p x^{q^p},$$

where $L_p \in GF(q^N)$ and $N(L)$ is the norm of the linearised polynomial. The norm $N(L)$ characterises the largest p where $L_p \neq 0$. Let $\otimes$ be the symbolic product of linearised polynomials defined as:

$$F(x) \otimes G(x) = F(G(x)) = \sum_{p=0}^{j} \sum_{i+l=p} \left(f_i g_l^{q^i} \right) x^{q^p},$$

where $j = N(F) + N(G)$. It is known that the symbolic product is associative and distributive, but it is non-commutative.

3 Berlekamp-Massey Algorithm for Rank-Codes

Now, we can describe the decoding algorithm for Rank-Codes with the modified Berlekamp-Massey algorithm. For simplicity we will define the minimum rank distance in the following sections by d. Let $\mathbf{c}$, $\mathbf{r}$, and $\mathbf{e}$ be the codeword vector, the received vector, and the error vector of length n with elements from $GF(q^N)$, respectively. The received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$. Let $v = r(\mathbf{e}|q)$ be the rank of the error vector $\mathbf{e}$. Now, we present a method of finding the correct codeword, if $2 \cdot v < d_r$.

We can calculate the syndrome $\mathbf{s}$ by:

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T. \tag{1}$$

Let us define a $(v \times n)$ matrix $\mathbf{Y}$ of rank v whose entries are from the base field $GF(q)$. Thus, we can write

$$\mathbf{e} = (E_0, E_1, \ldots, E_{v-1})\mathbf{Y}, \tag{2}$$

where $E_0, E_1, \ldots, E_{v-1} \in GF(q^N)$ are linearly independent over $GF(q)$. Now, we define the matrix $\mathbf{Z}$ as

$$\mathbf{Z}^T = \mathbf{Y}\mathbf{H}^T = \begin{bmatrix} z_0 & z_0^q & \cdots & z_0^{q^{d-2}} \\ z_1 & z_1^q & \cdots & z_1^{q^{d-2}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{v-1} & z_{v-1}^q & \cdots & z_{v-1}^{q^{d-2}} \end{bmatrix}. \tag{3}$$

It can be shown that the elements $z_0, z_1, \ldots, z_{v-1} \in GF(q^N)$ are linearly independent over $GF(q)$. Hence, equation (1) can be written as:

$$(S_0, S_1, \ldots, S_{d-2}) = (E_0, E_1, \ldots, E_{v-1}) \cdot \mathbf{Z}^T$$

or

$$S_p = \sum_{j=0}^{v-1} E_j z_j^{q^p} \quad , \quad p = 0, \ldots, d-2. \tag{4}$$

By raising each side of the equations to the power of q^{-p} we get:

$$S_p^{q^{-p}} = \sum_{j=0}^{v-1} E_j^{q^{-p}} z_j \quad , \quad p = 0, \ldots, d-2. \tag{5}$$

Hence, we have a system of $d-1$ equations with $2 \cdot v$ unknown variables that are linear in $z_0, z_1, \ldots, z_{v-1}$. Note that also the rank v of the error vector is unknown. It is sufficient to find one solution of the system because every solution of $E_0, E_1, \ldots, E_{v-1}$ and $z_0, z_1, \ldots, z_{v-1}$ results in the same error vector $\mathbf{e}$.

Let $\Lambda(x) = \sum_{j=0}^{v} \Lambda_j x^{q^j}$ be a linearised polynomial, which has all linear combinations of $E_0, E_1, \ldots, E_{v-1}$ over $GF(q)$ as its roots and $\Lambda_0 = 1$. We call $\Lambda(x)$ the row error polynomial. Also, let $S(x) = \sum_{j=0}^{d-2} S_j x^{q^j}$ be the linearised syndrome polynomial.

Now, it is possible to define the key equation by the next theorem.

Theorem 3 (Key Equation)

$$\Lambda(x) \otimes S(x) = F(x) \bmod x^{q^{d-1}}, \tag{6}$$

where $F(x)$ is an auxiliary linearised polynomial that has norm $N(F) < v$.

Proof 1 *From the definition of linearised polynomials we know that*

$$\Lambda(x) \otimes S(x) = \sum_{p=0}^{v+d-2} \left(\sum_{i+l=p} \Lambda_i S_l^{q^i} \right) x^{q^p}.$$

Since all coefficients $p \geq d-1$ vanish because of the modulo operation of equation (6) and the symbolic product of two linearised polynomials results in another linearised polynomial, we just have to prove that $F_p = 0$ for $v \leq p \leq d-2$.

$$\sum_{i+l=p} \Lambda_i S_l^{q^i} = \sum_{i=0}^{p} \Lambda_i S_{p-i}^{q^i} = \sum_{i=0}^{p} \Lambda_i \left(\sum_{s=0}^{v-1} E_s z_s^{q^{p-i}} \right)^{q^i}$$

$$= \sum_{s=0}^{v-1} z_s^{q^p} \left(\sum_{i=0}^{p} \Lambda_i E_s^{q^i} \right) = \sum_{s=0}^{v-1} z_s^{q^p} \Lambda(E_s) = 0$$

because p is equal to $v = N(\Lambda)$ or larger and $E_0, E_1, \ldots, E_{v-1}$ are roots of $\Lambda(x)$. □

Hence, we have to solve the following system of equations to get $\Lambda(x)$, if $2 \cdot v < d$,

$$-S_p = \sum_{i=1}^{v} \Lambda_i S_{p-i}^{q^i} \quad , \quad p = v, \ldots, 2v-1.$$

This can be written in matrix form as:

$$\mathbf{S} \begin{bmatrix} \Lambda_v \\ \Lambda_{v-1} \\ \Lambda_{v-2} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_v \\ -S_{v+1} \\ -S_{v+2} \\ \vdots \\ -S_{2v-1} \end{bmatrix}, \tag{7}$$

with $\mathbf{S}$ defined as:

$$\mathbf{S} = \begin{bmatrix} S_0^{q^v} & \cdots & S_{v-1}^{q^1} \\ S_1^{q^v} & \cdots & S_v^{q^1} \\ S_2^{q^v} & \cdots & S_{v+1}^{q^1} \\ \vdots & \ddots & \vdots \\ S_{v-1}^{q^v} & \cdots & S_{2v-2}^{q^1} \end{bmatrix}. \quad (8)$$

It can be shown that the matrix $\mathbf{S}$ is nonsingular. Thus, the system of equations has a unique solution. This solution can be efficiently found with a modified Berlekamp-Massey algorithm. This description of the Berlekamp-Massey algorithm is inspired by R. E. Blahut [13]. Equation (7) can also be seen as a feedback shift-register with tap weights given by $\Lambda(x)$. This is shown in figure 2. The symbols $f_1, f_2, \ldots, f_v$ stand for the powers of $q^1, q^2, \ldots, q^v$ (see equation (8)).

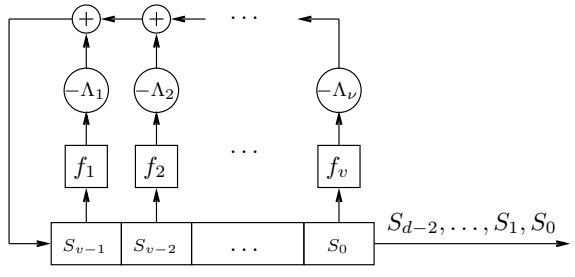


Fig. 2. Row error polynomial as a shift-register

The problem of solving the key equation is equivalent to a problem of finding the shortest feedback shift-register that generates the known sequence of syndromes. The design procedure is inductive. We start with iteration $r = 0$ and initialise the length of the shift-register $L_0 = 0$ and $\Lambda(x) = x$. For each iteration r we create a feedback shift-register that generates the first $r + 1$ syndromes and that has minimum length L_{r+1}. Hence, at the start of iteration r we have a shift-register given by $\Lambda^{(r)}(x)$ of length L_r. The notation of the exponent in brackets declares the iteration. To find $\Lambda^{(r+1)}(x)$ we determine the discrepancy of the output of the r-th shift-register and S_r by:

$$\Delta_r = S_r + \sum_{j=1}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j} = \sum_{j=0}^{L_r} \Lambda_j^{(r)} S_{r-j}^{q^j}. \quad (9)$$

For the case $\Delta_r = 0$, we set $\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x)$ and the iteration is complete. On the other hand, if $\Delta_r \neq 0$, the shift-register taps have to be modified in the following way:

Theorem 4 (Shift-Register Modification)
The linearised polynomial $\Lambda^{(r+1)}(x)$ is given by:

$$\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x) + A x^{q^l} \otimes \Lambda^{(m)}(x), \quad (10)$$

where $m < r$. Thus, if we choose $l = r - m$ and $A = -\Delta_r \Delta_m^{-q^l}$, the new discrepancy $\Delta'_r = 0$.

Proof 2 *From equation (9) it follows that*

$$\Delta'_r = \sum_{j=0}^{L_{r+1}} \Lambda_j^{(r+1)} S_{r-j}^{q^j}.$$

With equation (10) we can write:

$$\Delta'_r = \sum_{i=0}^{L_r} \Lambda_i^{(r)} S_{r-i}^{q^i} + A \sum_{i=0}^{L_m} \left(\Lambda_i^{(m)} S_{r-i-l}^{q^i} \right)^{q^l}$$

$$= \Delta_r + A \cdot \Delta_m^{q^l} = \Delta_r - \Delta_r \Delta_m^{-q^l} \cdot \Delta_m^{q^l} = 0,$$

where the syndrome $\mathbf{s}$ in the second sum has to be shifted for l positions because of the symbolic product with x^{q^l}. □

The new shift-register denoted by $\Lambda^{(r+1)}(x)$ has either length $L_{r+1} = L_r$ or $L_{r+1} = l + L_m$. It can be shown that we get a shortest shift-register for every iteration, if we choose m as the most recent iteration at which the shift-register length L_{m+1} has been increased. It was proved in [13] that the shortest feedback shift-register for Reed-Solomon codes in iteration r has length $L_{r+1} = \max\{L_r, r + 1 - L_r\}$. Furthermore, it is proved that the Berlekamp-Massey algorithm generates a shortest feedback shift-register in each iteration (see e.g. [12] or [14]). A similar proof as in [14] can be given for the modified Berlekamp-Massey algorithm of Rank-Codes.

Now, $\Lambda^{(r+1)}(x)$ generates the first $r + 1$ syndromes. This is shown in figure 3 for the case that the new shift-register has to be lengthened.

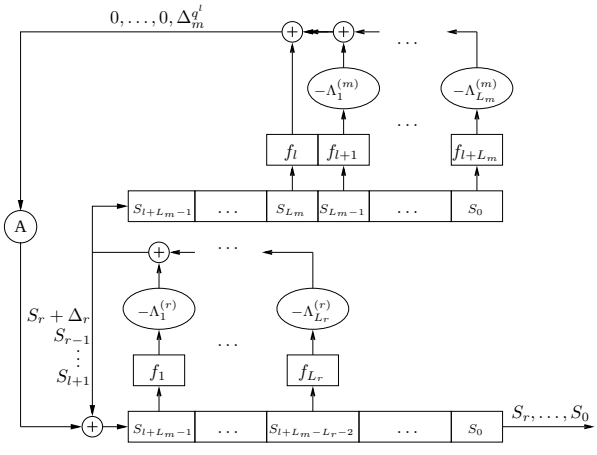


Fig. 3. Summation of the shift-registers

The shift-register of iteration m produces zeros at the first $m - 1$ iterations because there is an additional tap with weight one. At iteration m the shift-register produces $\Delta_m^{q^l}$ which is multiplicated by $A = -\Delta_r \Delta_m^{-q^l}$. This compensates Δ_r that was produced by the shift-register of iteration r. Hence, the new shift-register generates the sequence $S_0, S_1, \ldots, S_r$.

The modified Berlekamp-Massey algorithm for Rank-Codes can now be given as a flowchart. This

is shown in figure 4. $B(x)$ is an auxiliary linearised polynomial that is used to store $\Lambda^{(m)}(x)$, the row error polynomial of iteration m.

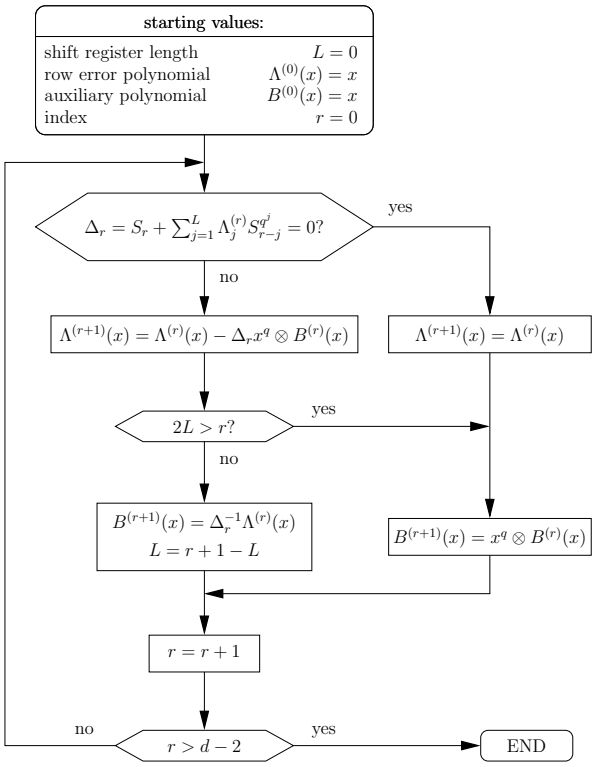


Fig. 4. Berlekamp-Massey algorithm for rank errors

Now, we can summarise the different steps of the decoding procedure.

1) Calculate the syndrome with equation (1).
2) Solve the key equation (7) with the modified Berlekamp-Massey algorithm to obtain $\Lambda(x)$.
3) Calculate the linearly independent roots $E_0, E_1, \ldots, E_{v-1}$ of $\Lambda(x)$. This can be done with the algorithm described in [11].
4) Solve the linear system of equations (5) for the unknown variables $z_0, z_1, \ldots, z_{v-1}$.
5) Calculate the matrix $\mathbf{Y}$ using equation (3).
6) Calculate the error vector $\mathbf{e}$ by equation (2) and the decoded codeword $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}$.

4 Berlekamp-Massey Algorithm for Rank-Codes with Row Erasures

In some applications we can get an information, which of the rows or columns are corrupted. Then we can declare the erroneous rows and columns as row and column erasures, respectively. In such a case, the decoder has to correct erasures and errors. Hereby, the location of the erasures are known but not the values of the located symbols. In this section we give a decoding method for row erasures and rank errors. The number of row erasures is s_r and the rank of the error matrix whose erased rows are filled with zeros is b. We can find the correct codeword, if $s_r + 2b < d$. Let $\Psi(x)$ be a linearised polynomial that has all linear combinations of the row erasures as its roots and $\Psi_0 = 1$. $\Psi(x)$ is called the row erasure polynomial.

The erroneous rows of the received codeword can be represented by $\Psi(x)$ as shown in the following example.

Example 2 (Row Erasures) Let the code $\mathcal{C}$ be in the field $GF(2^4)$ with the primitive polynomial $x^4 + x + 1$ and a codeword is

$$\mathbf{C} = [\alpha^0 \quad \alpha^{10} \quad \alpha^2 \quad \alpha^5] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Now, the codeword is interfered by erasures of the second and the fourth row. Thus, the received word is

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 1 & 1 & 1 \\ ? & ? & ? & ? \end{bmatrix} \begin{matrix} \leftarrow \alpha^0 \\ \leftarrow \alpha^1 \\ \leftarrow \alpha^2 \\ \leftarrow \alpha^3 \end{matrix}.$$

Hence, we set $E_0 = \alpha^1$ and $E_1 = \alpha^3$. The linear combinations of E_0 and E_1 determine the row erasure polynomial $\Psi(x) = x(x-\alpha^1)(x-\alpha^3)(x-(\alpha^1+\alpha^3)) = \alpha^2 x^4 + \alpha^9 x^2 + x$. ◇

Let $\nu = s_r + b$ and let $s_r + 2b < d$. We define $\tilde{\Lambda}(x)$ as the the row errata polynomial that has all linear combinations of $E_0, E_1, \ldots, E_{\nu-1}$ as roots. Without loss of generality we set $E_0, E_1, \ldots, E_{s_r-1}$ as the row erasures and $E_{s_r}, E_{s_r+1}, \ldots, E_{\nu-1}$ as the rank errors. The modified key equation (6) for the case with row erasures can be written as:

$$\tilde{\Lambda}(x) \otimes S(x) = F(x) \bmod x^{q^d - 1}, \qquad (11)$$

where the norm $N(F) < \nu$. The proof is similar to the proof in section 3. Again, we can represent equation (11) in matrix form:

$$\mathbf{S}_r \begin{bmatrix} \tilde{\Lambda}_\nu \\ \tilde{\Lambda}_{\nu-1} \\ \vdots \\ \tilde{\Lambda}_1 \end{bmatrix} = \begin{bmatrix} -S_\nu \\ -S_{\nu+1} \\ \vdots \\ -S_{b+\nu-1} \end{bmatrix}, \qquad (12)$$

with $\mathbf{S}_r$ defined as:

$$\mathbf{S}_r = \begin{bmatrix} S_0^{q^\nu} & S_1^{q^{\nu-1}} & \cdots & S_{\nu-1}^{q^1} \\ S_1^{q^\nu} & S_2^{q^{\nu-1}} & \cdots & S_\nu^{q^1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{b-1}^{q^\nu} & S_b^{q^{\nu-1}} & \cdots & S_{b+\nu-2}^{q^1} \end{bmatrix}.$$

Additionally, we know that the row erasures $E_0, E_1, \ldots, E_{s_r-1}$ are roots of $\tilde{\Lambda}(x)$. Hence, we

get the following system of equations:

$$\begin{bmatrix} E_0^{q^\nu} & \cdots & E_0^{q^1} \\ E_1^{q^\nu} & \cdots & E_1^{q^1} \\ \vdots & \ddots & \vdots \\ E_{s_r-1}^{q^\nu} & \cdots & E_{s_r-1}^{q^1} \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_\nu \\ \tilde{\Lambda}_{\nu-1} \\ \vdots \\ \tilde{\Lambda}_1 \end{bmatrix} = \begin{bmatrix} E_0 \\ E_1 \\ \vdots \\ E_{s_r-1} \end{bmatrix}. \quad (13)$$

It is possible to show that the $s_r + b$ equations for the unknown variables $\tilde{\Lambda}_1, \tilde{\Lambda}_2, \ldots, \tilde{\Lambda}_\nu$ are linearly independent and therefore have a unique solution. Also it is possible to show that the Berlekamp-Massey algorithm (see section 3) solves this system, if r and L_{s_r} are both initialised with s_r and the row errata polynomial $\tilde{\Lambda}^{(s_r)}(x)$ is initialised with $\Psi(x)$.

The initialisation of $\Psi(x)$ can be done iteratively. Let $E_0, E_1, \ldots, E_{s_r-1}$ be the s_r erasures and initialise $\Psi^{(0)}(x)$ with x. After iteration r we have $\Psi^{(r)}(x)$, which has all linear combinations of $E_0, E_1, \ldots, E_{r-1}$ as its roots. To get $\Psi^{(r+1)}(x)$ we calculate the r-th discrepancy Δ_r by:

$$\Delta_r = \Psi^{(r)}(E_r) = \sum_{j=0}^{r} \Psi_j^{(r)} E_r^{q^j}.$$

We can determine $\Psi^{(r+1)}(x)$ by:

$$\Psi^{(r+1)}(x) = \Psi^{(r)}(x) - \Delta_r^{-q+1} x^q \otimes \Psi^{(r)}(x).$$

The new discrepancy $\Delta_r' = 0$ because:

$$\Delta_r' = \Psi^{(r+1)}(E_r) = \Delta_r - \Delta_r^{-q+1} \cdot \Delta_r^q = 0.$$

Since the symbolic product by x^q does not change the roots of $\Psi(x)$, the new linearised polynomial $\Psi^{(r+1)}(x)$ has all linear combinations of $E_0, E_1, \ldots, E_r$ as its roots. Hence, the flowchart of the Berlekamp-Massey algorithm can be modified to correct row erasures and rank errors. This is shown in figure 5.

The decoding procedure of Rank-Codes with row erasures can be summarised as follows.

1) Fill the erased rows of the codeword with zeros or any other symbols and calculate the syndrome by equation (1).
2) Solve the key equation (12) and equation (13) with the modified Berlekamp-Massey algorithm for row erasures (see figure 5), where $E_0, E_1, \ldots, E_{s_r-1}$ have the values as shown in example 2.

Steps 3-7 are equivalent to the decoding algorithm described in section 3.

5 Decoding of Rank-Codes with Crisscross Erasures

The most difficult situation arises when the received codeword has erased rows and erased columns. Hence, in this section we give a decoding algorithm that deals with crisscross erasures and rank errors. The decoding algorithm is similar to [10]. Let s_r and s_c be the number

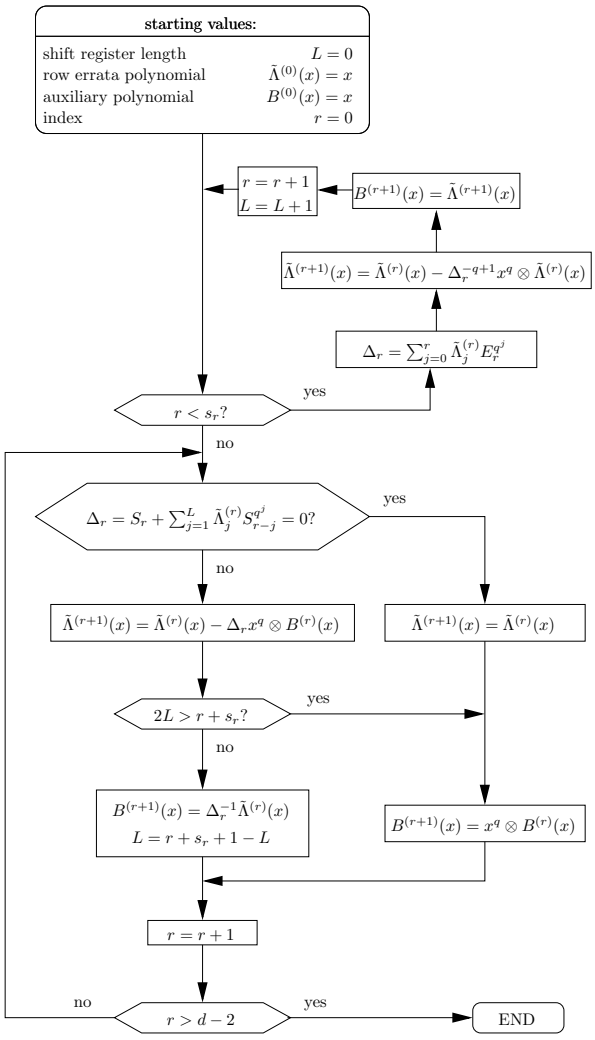


Fig. 5. Berlekamp-Massey algorithm for row erasures and rank errors

of row erasures and column erasures, respectively and let b now be the rank of the error matrix whose erased rows and erased columns are filled with zeros. The decoder will find the correct codeword when $s_r + s_c + 2b < d$. The decoding of Rank-Codes with crisscross erasures has to be done in two parts. The first part consists of the decoding of the punctured code. The second part is the recovering of the erased columns. The puncturing of the code is shown in the following example. The index p stands for the punctured code.

Example 3 Assume that a code $\mathcal{C}_{\mathcal{MRD}}(4, 2, 3)$ over the field $GF(2^4)$ is defined by the generator matrix

$$\mathbf{G} = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 \\ g_1^2 & g_2^2 & g_3^2 & g_4^2 \end{bmatrix}$$

and that the received array is.

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & ? & 0 \\ 0 & 1 & ? & 1 \\ 0 & 1 & ? & 1 \\ 1 & 0 & ? & 0 \end{bmatrix}.$$

The erased column must be deleted so that the punctured received array is:

$$\mathbf{R}_p = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The punctured generator matrix $\mathbf{G}_p$ that creates the punctured $\mathcal{C}_{\mathcal{MRD}}(3,2,2)$ code can be written as:

$$\mathbf{G}_p = \begin{bmatrix} g_1 & g_2 & g_4 \\ g_1^2 & g_2^2 & g_4^2 \end{bmatrix}.$$

$\diamond$

The first part consists of the decoding of the punctured code as described in section 4. Therefore, the punctured parity-check matrix has to be calculated by:

$$\mathbf{G}_p \cdot \mathbf{H}_p^T = 0. \tag{14}$$

The second part consists of the recovering of the erased columns as shown in the following example.

Example 4 Let a codeword array $\mathbf{R}$ has been decoded in the punctured code and let the array be:

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & ? & 0 \\ 0 & 1 & ? & 1 \\ 0 & 1 & ? & 1 \\ 1 & 0 & ? & 0 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ h_0 & h_1 & h_2 & h_3 \end{bmatrix}.$$

Since the variables $z_0, z_1, \ldots, z_{s_c-1}$ represent $h_0, h_1, \ldots, h_{n-1}$ we set $z_0 = h_2$. $\diamond$

After this we can calculate the values of $E_0, E_1, \ldots, E_{s_c-1}$ by equation (4) and continue with step 5 of the algorithm described in section 3.

The decoding procedure for crisscross erasures can now be summarised as follows.

1) Puncture the code as described in example 3 and calculate the punctured parity-check matrix by equation (14).
2) Start the decoding procedure of section 4 for the punctured code.
3) Fill the erased columns of the codeword with zeros or any other symbols and calculate the syndrome by equation (1).
4) Solve the linear system of equations (4) for the unknown variables $E_0, E_1, \ldots, E_{s_c-1}$, where $z_0, z_1, \ldots, z_{s_c-1}$ are the values of h_i of the erased columns (see example 4).
5) Continue with step 5 of the decoding procedure described in section 3.

6 Conclusions

We presented a modified Berlekamp-Massey algorithm for the decoding of codes with maximum rank distance, so-called Rank-Codes. We extended the modified Berlekamp-Massey algorithm for the decoding of row erasures and rank errors. At the end of this paper we gave a method for decoding Rank-Codes with crisscross erasures and rank errors. This algorithm which contains the decoding of the punctured code and the recovering of the erased columns is about twice as complex as the decoding algorithm for row erasures and rank errors. Therefore, we want to investigate a decoding method for column erasures and rank errors with less complexity in the future.

7 Acknowledgement

First of all we want to thank Prof. Ernst M. Gabidulin for some fruitful discussions and his hospitality in Moscow. Next, we want to thank Georg Schmidt for some helpful comments on this paper and Martin Bossert who led us to this topic.

References

[1] L. Levine and W. Meyers. Semiconductor memory reliability with error detecting and correcting codes. *Computers*, 9:43–50, October 1976.

[2] S.A. Elkind and D. P. Siewoiorek. Reliability and performance of error-correcting memory and register codes. *IEEE Trans. on Computers*, C-29(10):920–927, October 1980.

[3] W. F. Mikhail, R. W. Bartoldus, and R. A. Rutledge. The reliability of memory with single-error correction. *IEEE Trans. on Computers*, C-31(6):560–564, June 1983.

[4] A. M. Patel and S. J. Hong. Optimal rectangular code for high density magnetic tapes. *IBM J. Res. Dev.*, 18:579–588, November 1974.

[5] P. Prunsinkiewicz and S. Budkowski. A double track error-correction code for magnetic tape. *IEEE Trans. on Computers*, C-25(6):642–645, June 1976.

[6] M. Blaum and R. J. McEliece. Coding protection for magnetic tapes: A generalization of the patel-hong code. *IEEE Trans. Inf. Theory*, IT-31(5):690–693, September 1985.

[7] P. Delsarte. Bilinear forms over a finite field with applications to coding theory. *Journal of combinatorial theory. Series A*, 25(4):226–241, 1978.

[8] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, Januar–March 1985.

[9] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE Trans. Inf. Theory*, 37(2):328–336, March 1991.

[10] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Rank errors and rank erasures correction. *Colloquium on Coding Theory*, pages 11–19, 1992.

[11] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw Hill, New York: Mc Graw-Hill, 1968.

[12] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inf. Theory*, IT-15:122–127, January 1969.

[13] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, Owego, New York 13827, 1983. ISBN 0-201-10102-5.

[14] K. Imamura and W. Yoshida. A simple derivation of the Berlekamp-Massey algorithm and some applications. *IEEE Trans. Inf. Theory*, IT-33:146–150, January 1987.